Project Overview | Coursera

coursera.org/learn/security-with-aws-identity-and-access-management-iam-dvoa0/supplement/SfTtV/project-overview

Welcome to Your Guided Project!

Welcome to **Security with AWS: Identity and Access Management (IAM)**. This intermediate project-based course should take approximately 2-and-a-half hours to finish. Before diving into the project, please take a look at the course objectives and structure:

Guided Project Objectives

In this course, we are going to focus on **three** learning objectives:

1. Set up secure user access in AWS IAM, including creating and managing user accounts, groups, and policies

2. Set up secure user access and manage permissions to create roles for cross-account access and implement best practices

3. Manage permissions within AWS IAM by configuring policies, recognizing policy structures, and using condition keys to enforce granular access controls.

By the end of this project-based course, you will be able to secure user access, manage permissions, create roles for cross-account access, and implement IAM best practices. We will introduce core AWS IAM concepts and principles and grant secure access to a thirdparty auditing firm to review financial reports stored in another AWS account.

Guided Project Structure

This course is divided into **six (6)** parts:

1. Project Overview: This introductory reading material.
2. Reading: Link to project resources
3. Reading: Links to additional project resources
4. **Security with AWS: Identity and Access Management (IAM):** This is the hands-on project that we will work on together in Rhyme
5. Graded Quiz: This is the final assignment you must pass to finish the project successfully.
6. **Course End (Learner) Survey:** Tell us what you thought about this guided project!

Project Structure

The hands-on project on **Security with AWS: Identity and Access Management (IAM)** is divided into the following tasks:

Task 1: Set up and overview of the project

- Overview of the project
- A brief introduction to the Rhyme platform
- Set up AWS MFA as a security best practice for root and IAM users

Task 2: Create an IAM user from the console

- Create an IAM user called **SecurityTeamAdmin** with admin privileges
- Login to the AWS management console as the **SecurityTeamAdmin**

Task 3: Create IAM users from the CLI

- Generate access keys from the AWS **SecurityTeamAdmin's** account
- Configure the command line interface (CLI) to the **SecurityTeamAdmin's** account using the access keys

- Create three IAM users (Matt, Sarah, and Deborah) from the CLI

Practice Activity: Create an IAM user from the CLI

Create an IAM user called **Rachel** from the CLI.

Task 4: Create IAM groups and add users to the groups

- Create two groups, one from the console and the other from the CLI: **AdminGroup** and **CloudSecurityTeam**.
- Add IAM users (Deborah and SecurityTeamAdmin) to the **AdminGroup** and attach the **AWSAccountManagementFullAccess** policy to the group from the console.
- Add IAM users (Matt and Sarah) to the **CloudSecurityTeam** and attach the **Amazons3FullAccess** policy to the group from the CLI.

Practice Assessment: Securing AWS accounts

Answer two quiz questions on securing AWS accounts.

Task 5: Implementing IAM policies

- Introduce IAM policy concepts
- Create a customer-managed policy called **IAMReadPolicy** that enables read permissions for IAM components.

Task 6: Create and upload to an S3 bucket

Create an AWS S3 bucket and upload two files into the bucket

Task 7: Create an IAM role for an AWS service

- Introduce the concept of AWS IAM roles
- Create an IAM role called **EC2toS3Role** that allows an AWS service - EC2, to communicate with another AWS service - S3.
- Launch an EC2 instance with default settings and a name as **List S3 buckets**.
- Attach the IAM role **EC2toS3Role** to the EC2 instance
- Connect the instance via the **EC2 Instance Connect** and retrieve the instance metadata, including the security credentials.

Practice Activity: Create a customer-managed policy

Create a customer-managed policy called **S3ListAndReadPolicy** that grants IAM users access to list and read S3 buckets.

Task 8: Use IAM roles to grant AWS cross-account access

- Log in to another AWS account as the **AuditTeamAdmin** using the provided credentials (username and password).
- Create an IAM role called **AuditFinData** that allows the audit team admin access to financial data by attaching the **AmazonS3ReadOnlyAccess** policy.
- Switch roles to the **SecurityTeamAdmin's** account from the **AuditTeamAdmin's account.**
- Confirm that the **AuditTeamAdmin** can only perform the actions granted by the **AmazonS3ReadOnlyAccess** policy.

Task 9: Use IAM roles to grant AWS cross-account access with external ID

- Configure the CLI as the **AuditTeamAdmin's** account using the credentials (access key ID and secret access keys) provided.
- Create an IAM role with an external ID called **AuditFinDataExtID** that grants **List** and **Read** S3 bucket permissions to the **AuditTeamAdmin** through the **S3ListAndRead** policy created earlier.
- Get temporary access to the **SecurityTeamAdmin's** account from the **AuditTeamAdmin's** account through the **aws sts assume-role** command on the CLI.
- Set the access key ID, secret access key, and session token generated by the STS.
- Confirm that the AuditTeamAdmin can only perform the actions granted by the **S3ListAndRead** policy**.**

Task 10: Revoke an IAM role

- Revoke the IAM role called **AuditFinDataExtID**, whose security credentials may have been leaked.
- Confirm that S3 access is denied after the role was successfully revoked.

Practice Activity: Get cross-account access after revoking a role

- Close the command line and reopen the command line.
- Reconfigure the command line to the audit team admin's account using aws configure.
- Generate and set new access key ID, secret key, and session ID.
- Close the command line and reopen the command line.
- Confirm that you can list the S3 bucket in the security team's account using **aws s3 ls.**

Task 11: Setting permissions boundary

- Illustrate how the permissions boundary works.
- Create a customer-managed policy called **IAMPermissionBoundary** that will restrict an IAM user from changing his permissions and creating another user with full administrator access.
- Create the IAM user James with full IAM admin privileges and enforce the permissions boundary.
- Login to James' AWS account to see that the permission boundary works.

Task 12: Test and Debug IAM policies using the IAM policy simulator

- Create a customer-managed policy called **DenyS3CreateBucket** that denies access to create an S3 bucket and attach the policy to an IAM user, **Matt**.
- Open the IAM policy simulator console and simulate for different allow and deny permissions.
- Wrap up the project.

Cumulative Activity: Deal with a cloud security breach

Create a role granting permissions to AWS services (an EC2 instance) to communicate with S3, launch an EC2 instance, get the security credentials of that role, revoke the role, and terminate the EC2 instance.

Meet the Instructor

My name is Olayinka Imisioluwa Arimoro, and I will be taking you through this Guided Project. I am a guided project instructor interested in statistical and machine learning applications to human disease, clinical trials, and patient-reported outcomes. I am passionate about teaching others relevant skills in different sectors.

About Rhyme

This project runs on Coursera's hands-on platform called Rhyme. On Rhyme, you do projects in a hands-on manner in your browser. You will get instant access to pre-configured cloud desktops with all the necessary software and resources. So you can focus on learning. For this project, this means instant access to a cloud desktop with AWS CLI version 2 pre-installed.